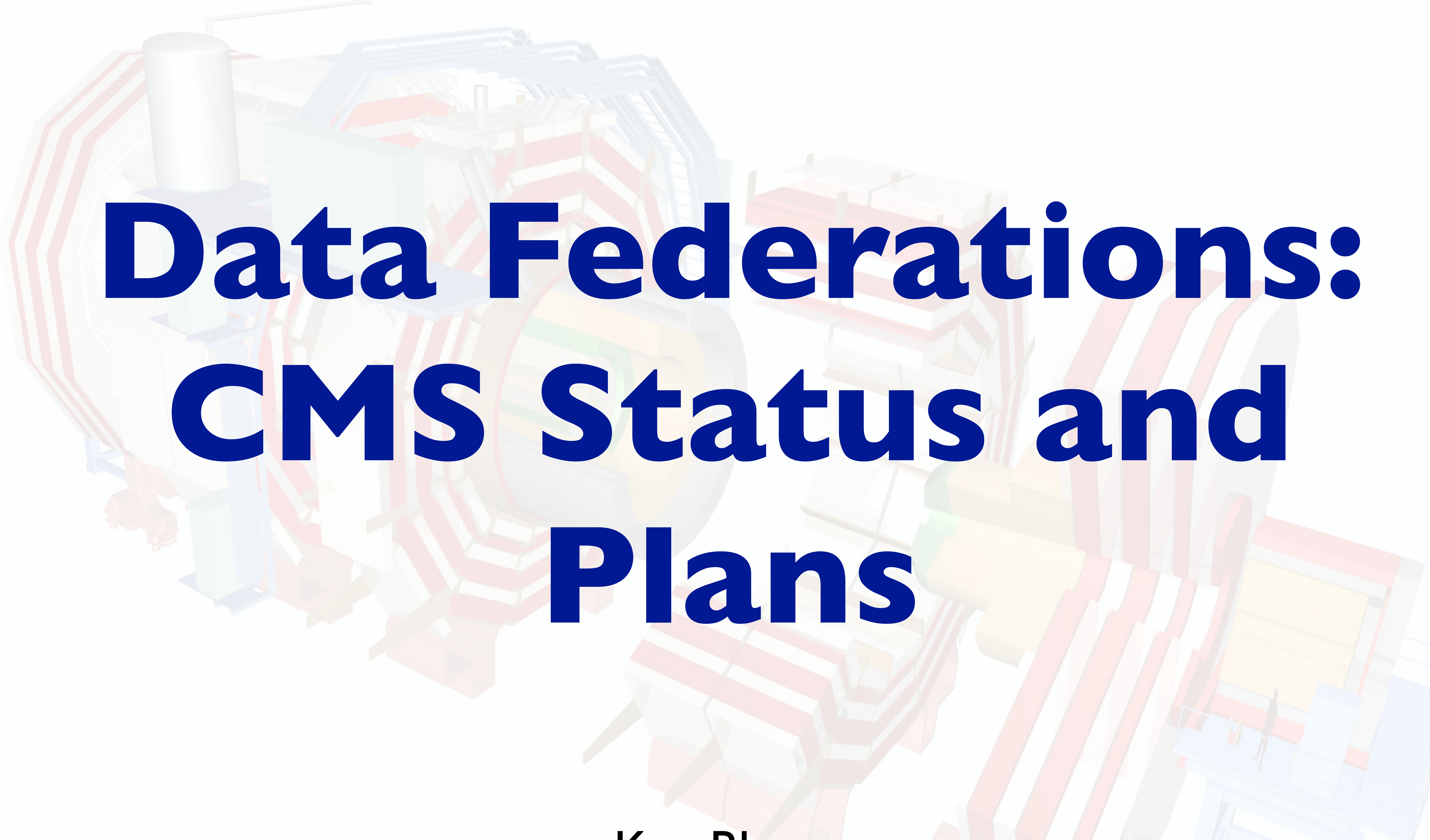
Data Federations: CMS Status and Plans



Ken Bloom
For the CMS, US CMS and AAA teams
April 10, 2014

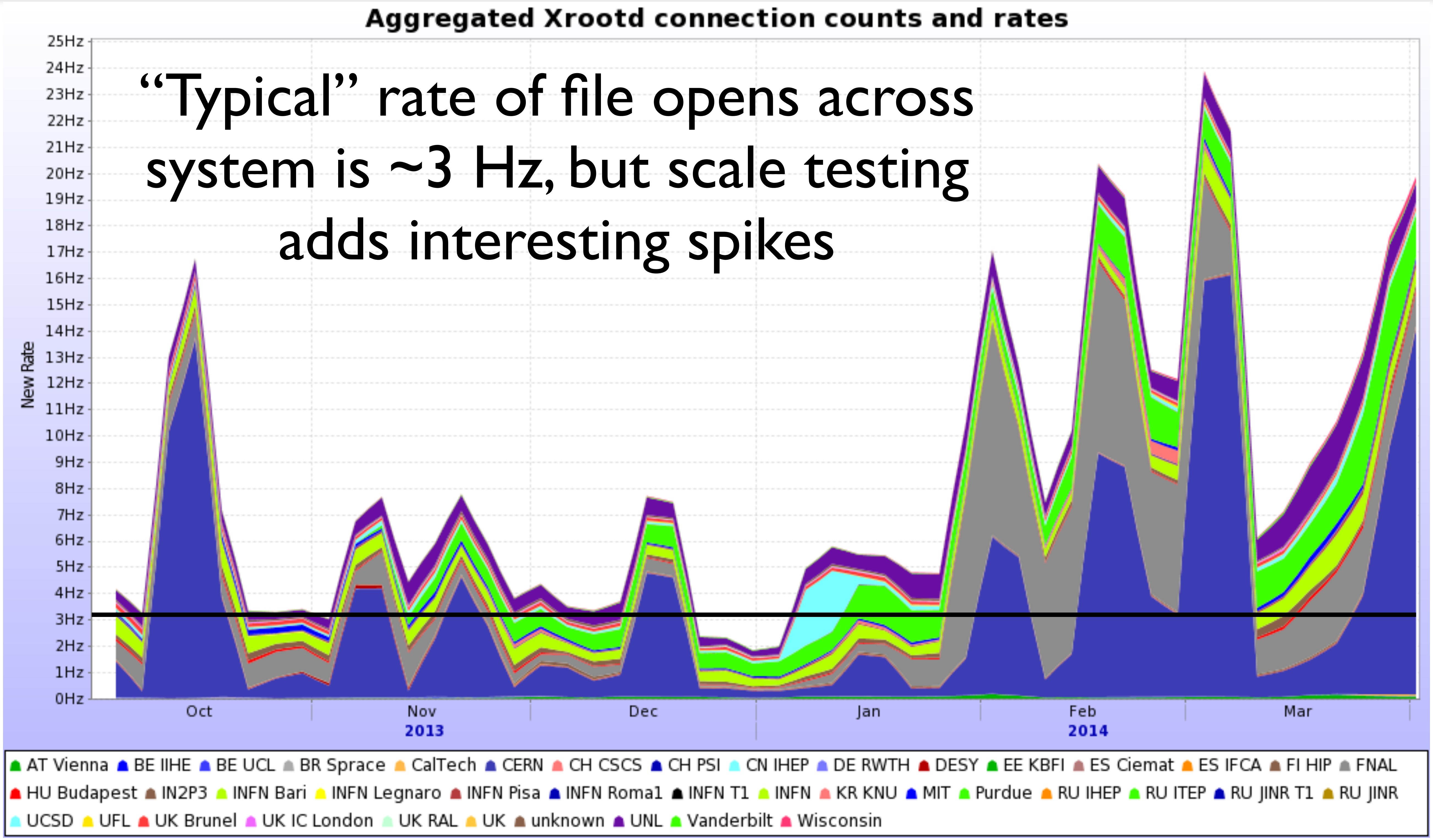
Some history

- I first heard the data federation idea in March 2010, when US CMS was in the final throes of the Hadoop revolution
- At that time, CMS software had poor I/O performance; a lot of work was done to improve that (see Maria's talk)
  - This is what makes wide-area data access functionally possible
- A year later (2011), we had understood the basic use cases, had four sites in US CMS behind a regional redirector, and were encouraging US T2 sites to configure fallback to WAN access
  - NSF-funded "Any Data, Anytime, Anywhere" grant started in 2011 at Nebraska, UCSD, Wisconsin
  - "AAA" is the CMS branding of our data-federation implementation
- In 2012, all US CMS sites were in the federation, with some European sites joining in through a redirector in Italy
- One year ago, there were 21 CMS T2 sites in the federation, and I think 2 T1's. We officially asked all sites to join. Today….

Status of AAA deployment

- 6 of "8" CMS T1 sites are part of the data federation
  - In: DE, FR, IT, (RU,) UK, US
  - Not in: ES (coming soon), TW ("opportunistic T1")
  - Important caveat: CMS T1 sites are in the midst of disk-tape separation, so that we have greater control over what files are currently on disk. Only the files on disk are actually accessible.
  - In principle this already gives access to a huge amount of CMS data
- 41 of 52 CMS T2 sites are part of the data federation
  - In general, the sites that are not in the federation (without naming names) are smaller and/or less robust
  - ~ 96% of unique datasets resident at T2's are available
- We consider this to be full deployment within CMS!

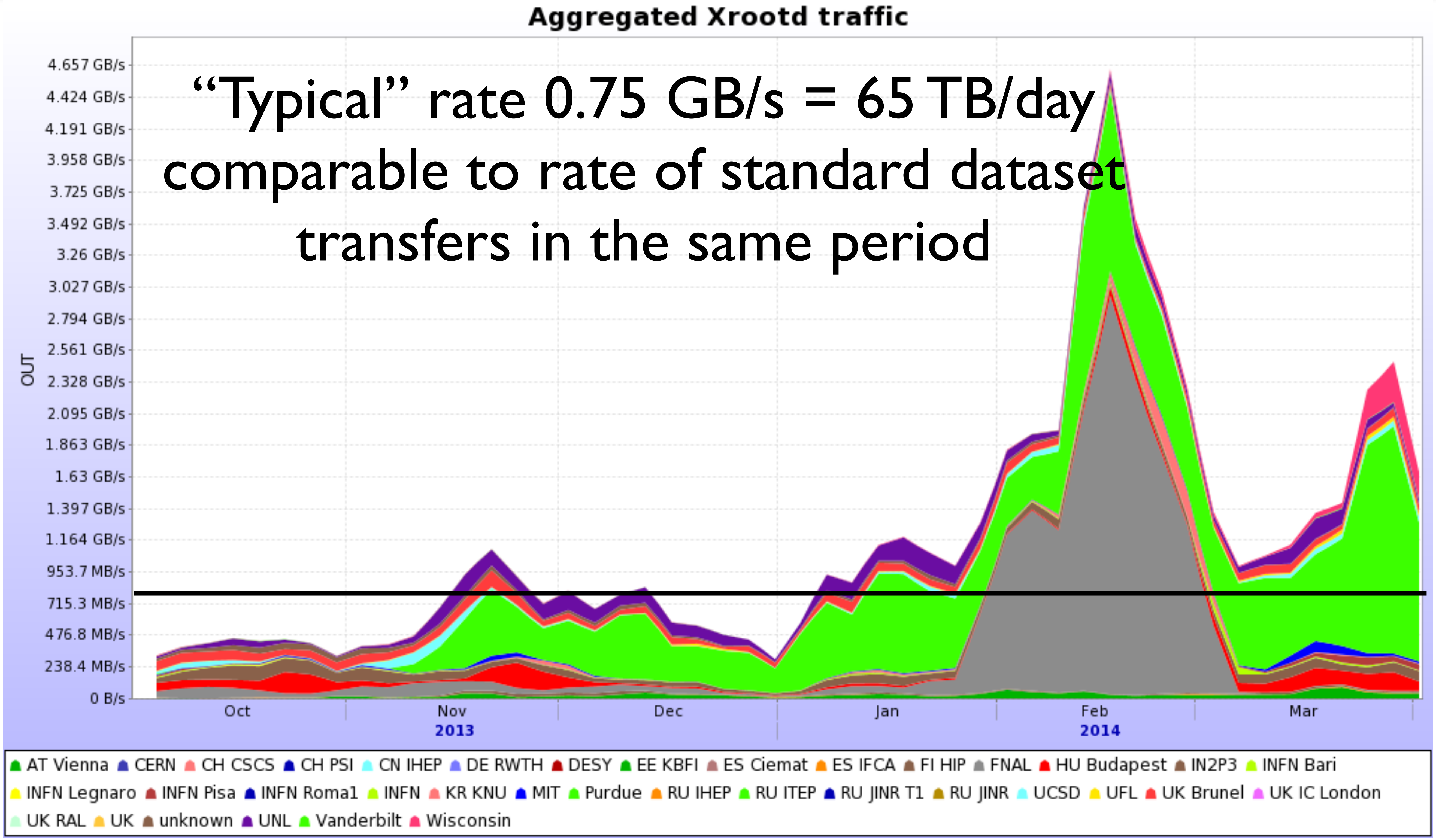
Global picture



"Typical" rate of file opens across system is ~3 Hz, but scale testing adds interesting spikes

http://xrootd.t2.ucsd.edu

Global picture



"Typical" rate 0.75 GB/s = 65 TB/day comparable to rate of standard dataset transfers in the same period

Note: results may be skewed by scale testing

Applications: fallback

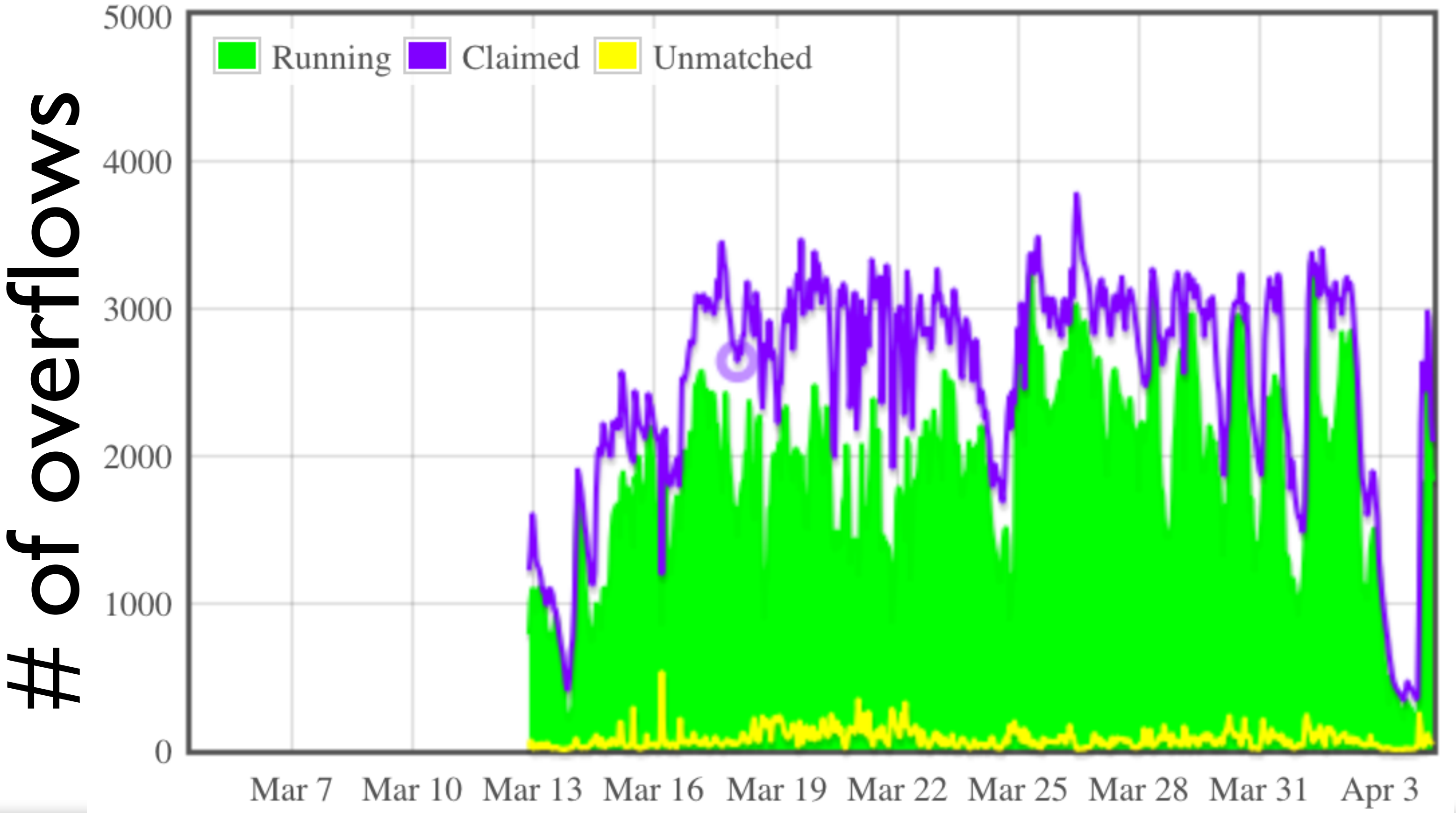
- One of the first applications of AAA was the “fallback mechanism”
  - This is the key to almost every other AAA application….

- Usually, if a job fails to open an input file, it crashes
- The fallback mechanism gives a path for recovery:
  - On file-open failure, CMSSW asks redirector to find file elsewhere
  - Job then reads remote file, user never notices
- More throughput for users, less CPU time wasted on failed jobs
- Makes entire system more robust against single-site storage issues

- A few easy configuration changes needed at sites to do this
  - 47/52 T2 sites have implemented fallback
  - One T1 has not due to firewall issues; discussions/debugging continue on proxy server deployment there

Applications: efficiency for users

- Sites with popular datasets can have very long batch queues
- Re-direct jobs to another site with free job slots, read data via AAA
  - Smaller CPU efficiency, but jobs can start sooner
  - Achieved by changing scheduling policies in glideinWMS layer, regulate number of jobs to match WAN bandwidth
- So far, only small scale -- overflow amongst four sites in the US, ~O(2K) simultaneous jobs -- but no technical issues block expansion

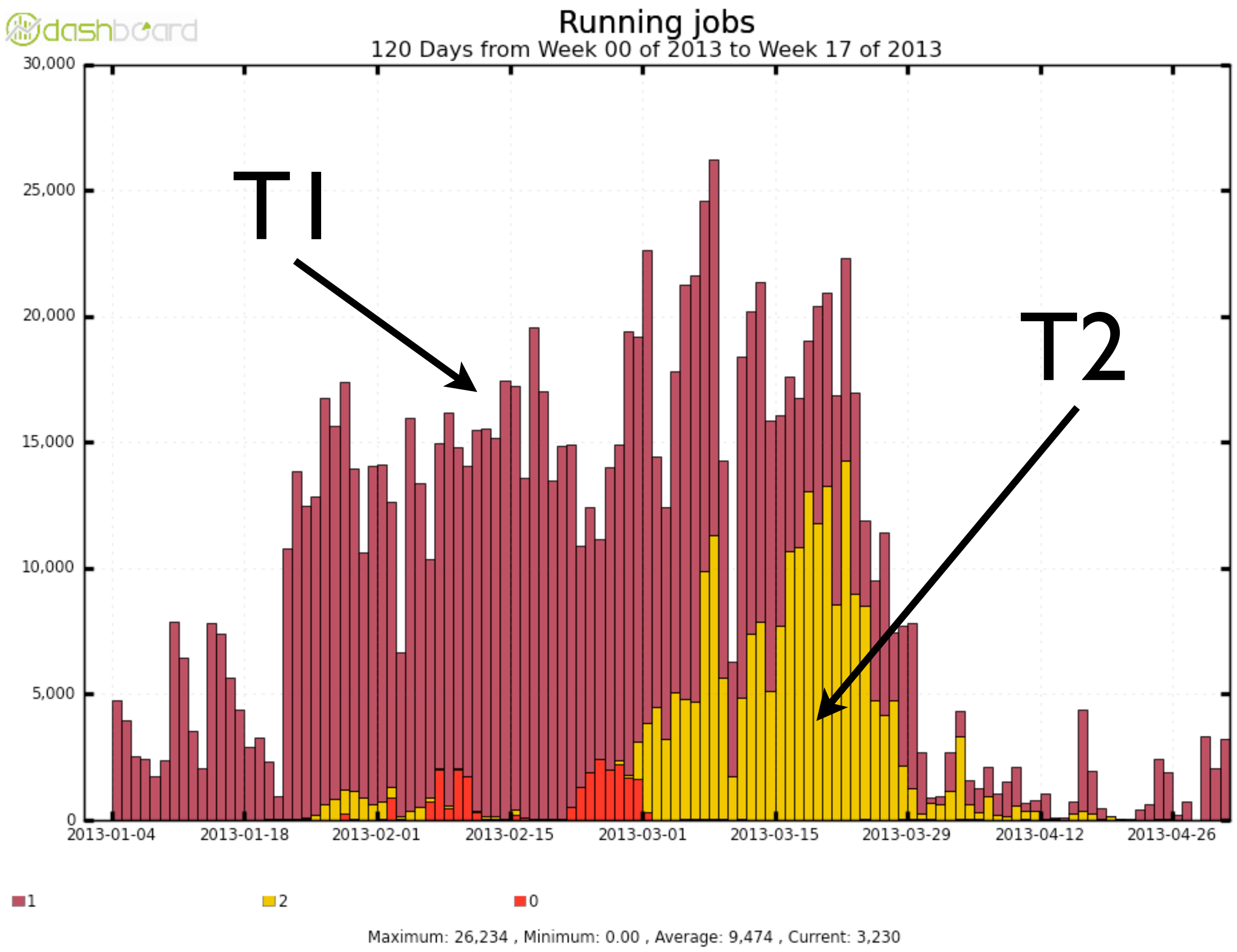


Applications: sites without data

- Some T3 sites are completing entire data analyses through AAA
  - Observed ~800 simultaneous jobs, 2-3 Gb/s WAN input sustained for a week, 99% success rate
  - Much satisfaction with local control over processing resources
  - "At this point, I basically don't pay attention to where the data is and just assume that jobs will find the data and run."
- Exploring possibility of diskless T2 sites at well-networked centers
- Sites that temporarily lose their data due to storage downtime (planned or unplanned) can continue to operate as normal through the fallback mechanism
  - Allows the continuity of processing capacity, system-wide
  - Have seen several successful cases, some planned and some not

Applications: production with remote data

- "Legacy" reprocessing of 2012 data and associated simulation samples
- Inputs resident at T1 sites
  - T1's ran on data locally
  - T2's ran on simulations read via AAA fallback mechanism
- Whole job done faster
- This gives us flexibility in workflow location that may be very much needed during the next LHC run
  - Already being put to use in idle HLT farm



Applications: opportunistic usage

- Any data, anywhere means any computer, not just CMS-owned
  - For software, use Parrot and CVMFS for download on demand, brings in 500 MB of files rather than 17 GB
  - Then, read data through AAA fallback mechanism
  - Typical jobs only 2% slower than those running on CMS sites
- Opens the door to any opportunistic resource, e.g. clouds
  - Have run 2K simultaneous jobs across 15 non-CMS OSG sites, including ATLAS sites (thanks)
  - Successful demonstration on Amazon cloud
  - Much CMS development work underway

What could hold us back?

- I see two categories of issues that could keep CMS from fulfilling the promise of all of the above applications:
- Technical: We encounter scaling problems on either the serving or hosting ends that lead us to enforce some kind of throttling
  - Carl Vuosalo will discuss scale tests later today
- Psychological/sociological: need to educate/convince users that AAA will work for them
  - Push user education efforts
  - CSA14, scheduled for this summer, is an opportunity to show off what can be achieved with data federations; will make sure that the system is thoroughly exercised

Requirements to join the federation

- There are no requirements for joining the federation beyond being able to follow the instructions for deploying xrootd
- Lesson learned from years of T2 coordination: it is extremely difficult to bring all sites up to some technical standard up front
  - Instead, be forward-leaning and try things, then sort out problems
  - Only had to kick one site out of the federation so far
- Currently running scale tests against every site in the federation to understand limitations
  - Information will be passed to operations team, which will use this to determine usage guidelines, site by site
  - But we will probably need some technical guidance from developers on how to best use xrootd with different backend storage systems

Tolerance

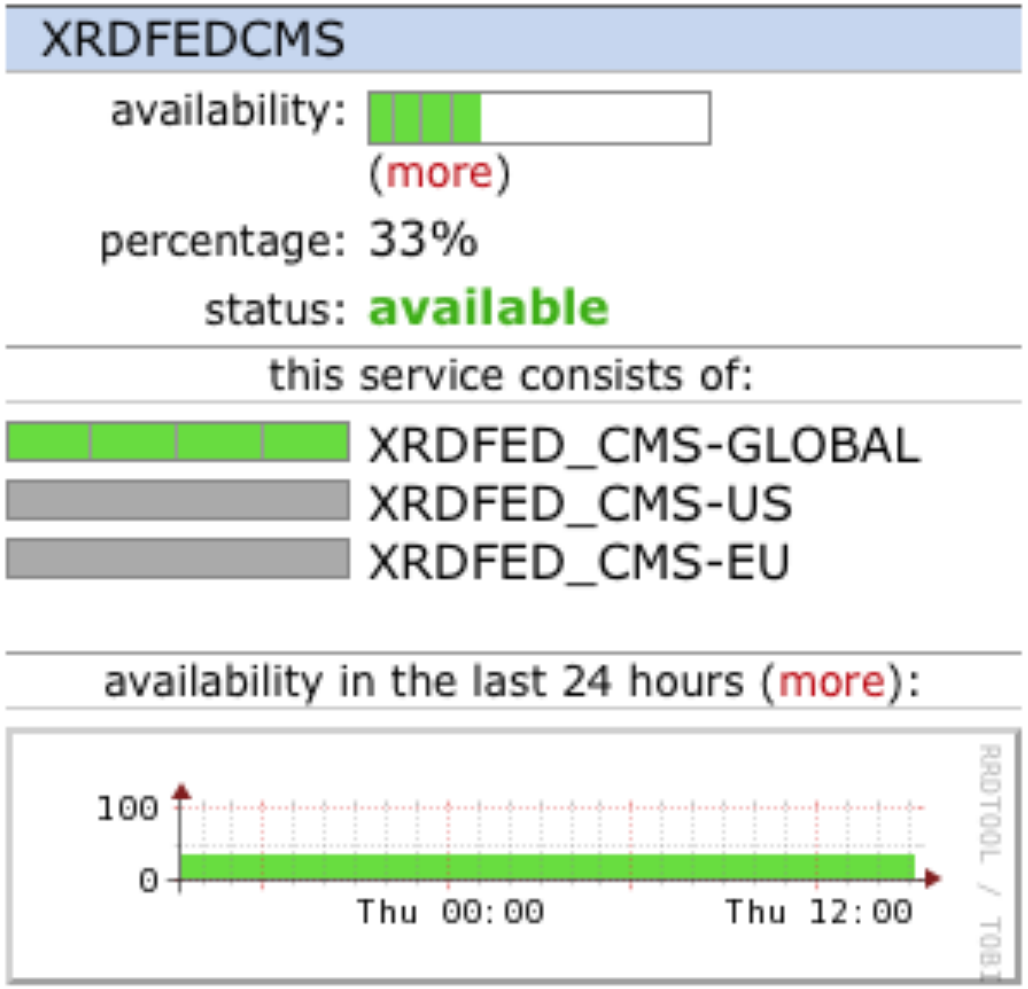
- Rather than try to establish fault-free sites, focus on building fault-tolerant systems
  - The fallback mechanism is already an example of this
- But fallback is considered successful as soon as another location for the file is found, even if the open attempt fails
  - Want to be able to transparently attempt to read from a different site instead (work in progress)
- Want to have "smart routing" that can be aware of poor network/ storage performance at sites and can adapt and recover on the fly
  - Can we be smart not just at file-open time but during file reads?

CMS-specific monitoring: sites

- Two SAM (site availability monitoring) tests:
  - Fallback: Can site successfully fall back when a file is missing locally?
  - Access: Can a file at the site be opened via AAA?
    - Makes use of the "TFC trick" to make a file appear to exist only at the one site in question
  - Each runs about once/hour
- Neither test is yet considered "critical" (required), but we want to make the fallback test so as soon as the firewall problem is resolved at the one T1 site
  - For operational purposes, fallback more important than access
- Failing of these tests: all sites "fail" if the central infrastructure fails

CMS-specific monitoring: infrastructure

- Starting to put some simple tests in the SLS infrastructure
  - Work in progress…
- Do simple functional tests of redirectors, make sure they are alive
- When ready, test status will be shown to shifters, who can then send tickets, alert experts, etc.



XRDFEDCMS

availability: 🟩🟩🟩
(more)

percentage: 33%

status: **available**

this service consists of:

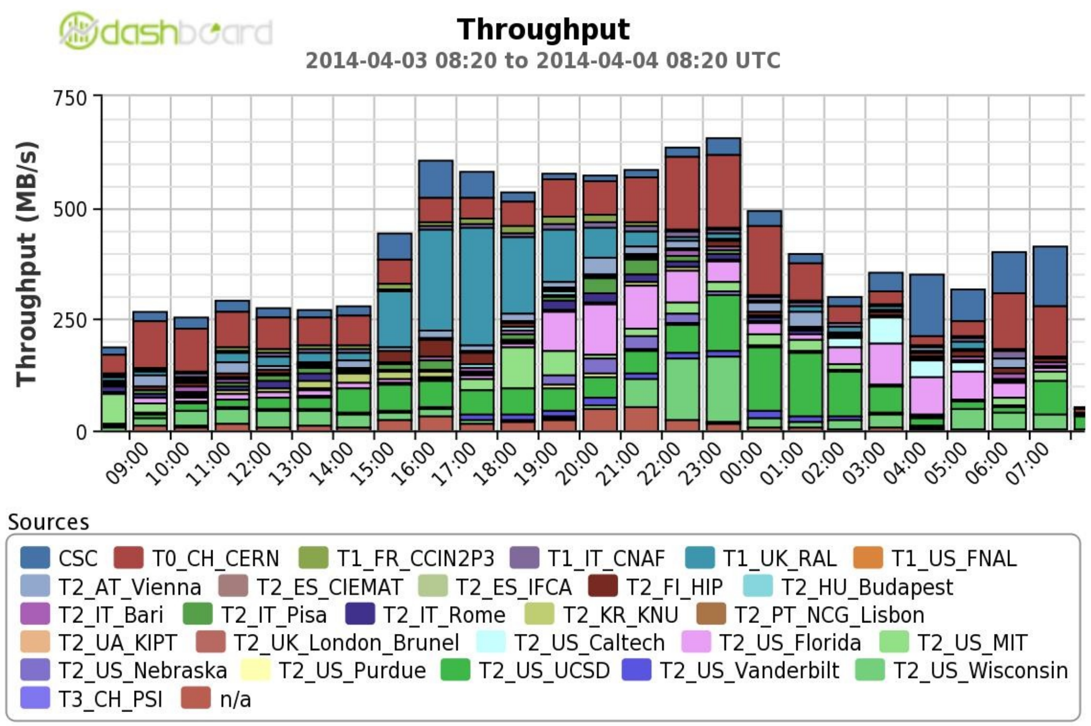
🟩 XRDFED_CMS-GLOBAL

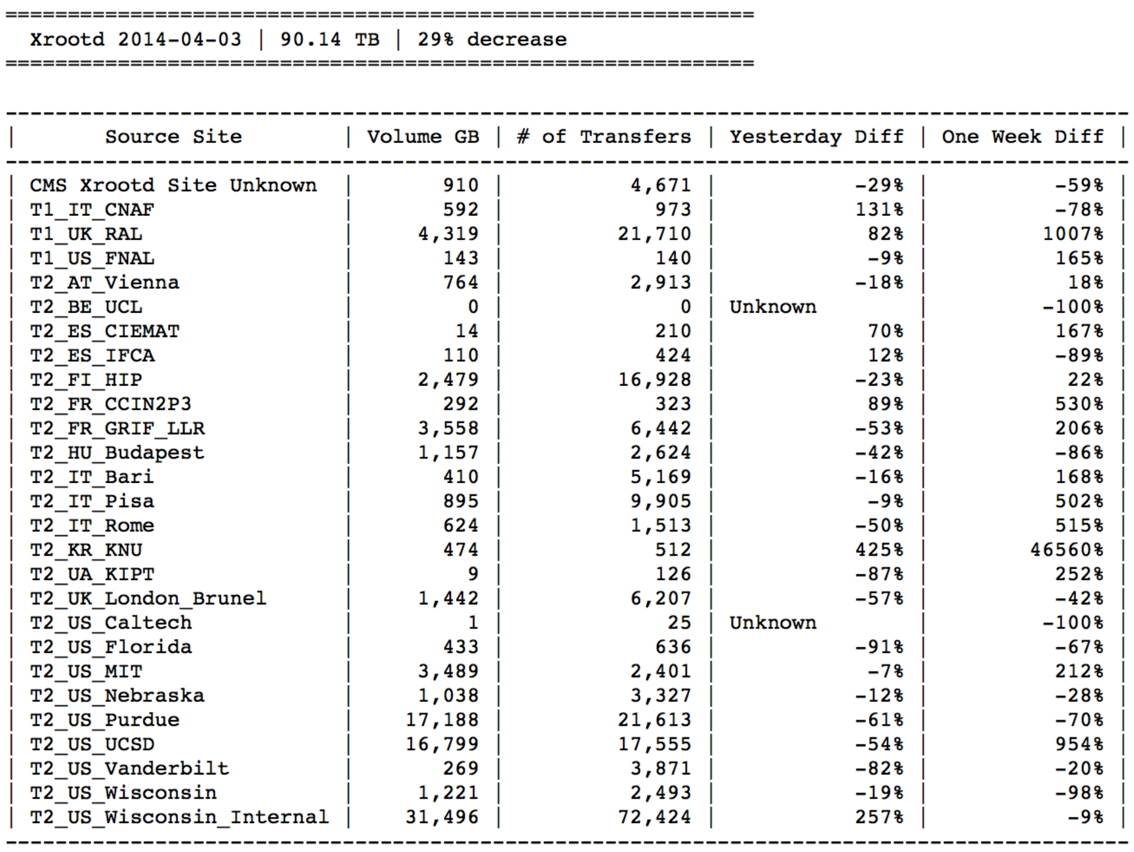
▨ XRDFED_CMS-US

▨ XRDFED_CMS-EU

availability in the last 24 hours (more):

CMS-specific monitoring: accounting

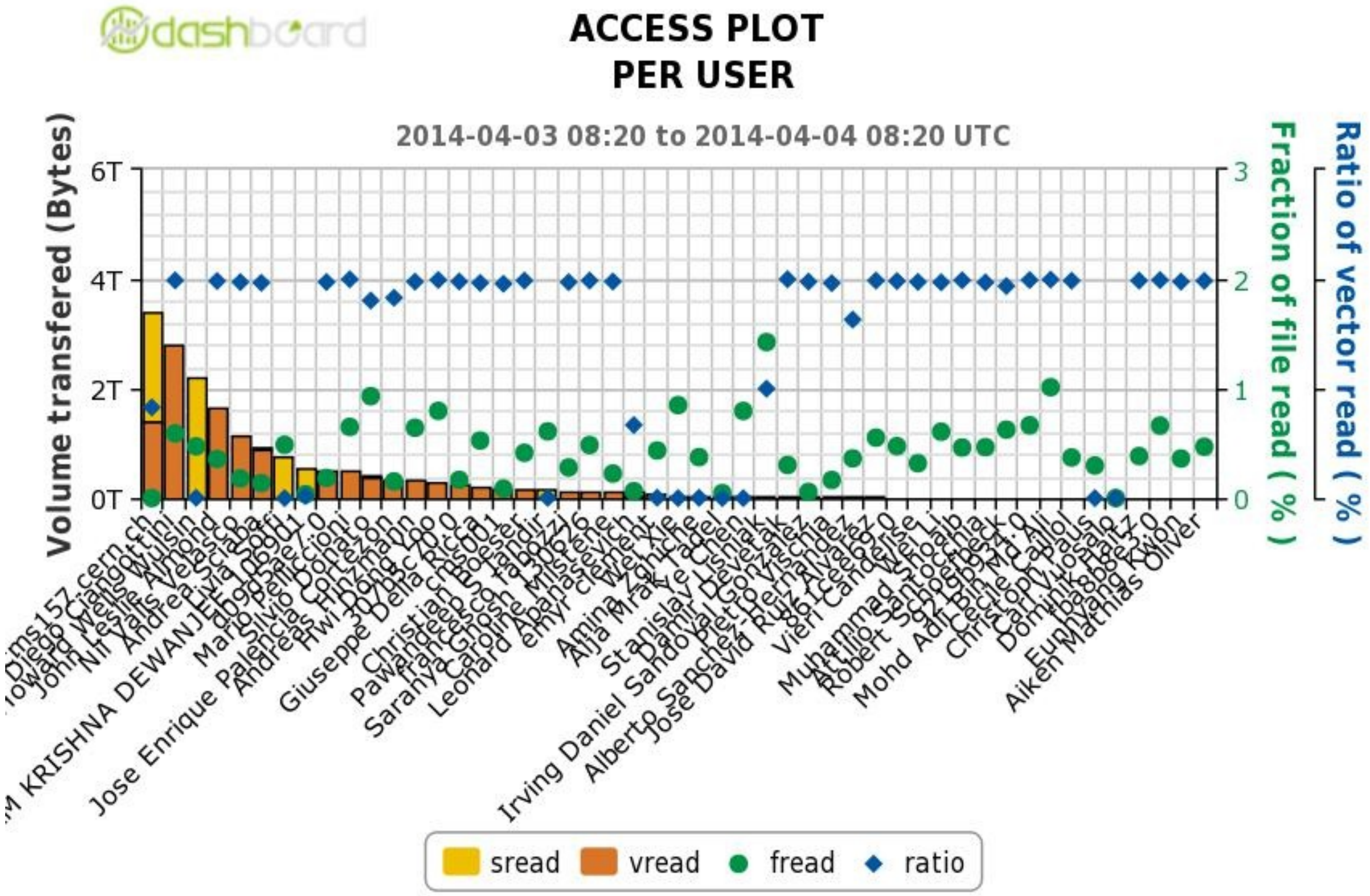




```
============================================================
 Xrootd 2014-04-03 | 90.14 TB | 29% decrease
============================================================

------------------------------------------------------------------------------------------
|        Source Site        | Volume GB | # of Transfers | Yesterday Diff | One Week Diff |
------------------------------------------------------------------------------------------
| CMS Xrootd Site Unknown   |       910 |          4,671 |           -29% |          -59% |
| T1_IT_CNAF                |       592 |            973 |           131% |          -78% |
| T1_UK_RAL                 |     4,319 |         21,710 |            82% |         1007% |
| T1_US_FNAL                |       143 |            140 |            -9% |          165% |
| T2_AT_Vienna              |       764 |          2,913 |           -18% |           18% |
| T2_BE_UCL                 |         0 |              0 | Unknown        |         -100% |
| T2_ES_CIEMAT              |        14 |            210 |            70% |          167% |
| T2_ES_IFCA                |       110 |            424 |            12% |          -89% |
| T2_FI_HIP                 |     2,479 |         16,928 |           -23% |           22% |
| T2_FR_CCIN2P3             |       292 |            323 |            89% |          530% |
| T2_FR_GRIF_LLR            |     3,558 |          6,442 |           -53% |          206% |
| T2_HU_Budapest            |     1,157 |          2,624 |           -42% |          -86% |
| T2_IT_Bari                |       410 |          5,169 |           -16% |          168% |
| T2_IT_Pisa                |       895 |          9,905 |            -9% |          502% |
| T2_IT_Rome                |       624 |          1,513 |           -50% |          515% |
| T2_KR_KNU                 |       474 |            512 |           425% |        46560% |
| T2_UA_KIPT                |         9 |            126 |           -87% |          252% |
| T2_UK_London_Brunel       |     1,442 |          6,207 |           -57% |          -42% |
| T2_US_Caltech             |         1 |             25 | Unknown        |         -100% |
| T2_US_Florida             |       433 |            636 |           -91% |          -67% |
| T2_US_MIT                 |     3,489 |          2,401 |            -7% |          212% |
| T2_US_Nebraska            |     1,038 |          3,327 |           -12% |          -28% |
| T2_US_Purdue              |    17,188 |         21,613 |           -61% |          -70% |
| T2_US_UCSD                |    16,799 |         17,555 |           -54% |          954% |
| T2_US_Vanderbilt          |       269 |          3,871 |           -82% |          -20% |
| T2_US_Wisconsin           |     1,221 |          2,493 |           -19% |          -98% |
| T2_US_Wisconsin_Internal  |    31,496 |         72,424 |           257% |           -9% |
------------------------------------------------------------------------------------------
```

- Daily emails to AAA team
- Many nice plots available from CERN IT dashboard



CMS-specific monitoring: issues

- Not all sites are currently reporting the detailed monitoring information — currently have 4 T1 sites, 21 T2 sites
- Most (but not all) of the missing sites are dCache sites
- At last check, there were difficulties with respect to those versions of dCache that worked correctly with the plugin, and those that supported SHA-2
- CMS would benefit from some more clear guidance on how to get all of the monitoring capabilities implemented for each type of storage technology that we run

My distributed storage vision

- AAA is now giving us excellent *read* access to federated data
- This would be more useful to users if they could *manage* their federated data as they do with data on a local disk
  - E.g. users could do directory listings across the federated namespace
  - Or, admins could easily measure total usage for each user, and perhaps impose a quota across the distributed storage
  - The CMS namespace is structured in a way that makes this possible, but we seem to lack the necessary technical tools
- With this greater functionality, more of the work of deploying/ operating user storage could be given to centrally operated facilities and support teams — easier and more cost efficient for participating physicists and sites
- (Yes, I am basically asking for Dropbox functionality)

Outlook

- The data federation, and its implementation through xrootd, has turned out to be a very nice fit with CMS
  - Thanks to robust WAN, straightforward namespace, I/O efforts
- Benefits from one user who wants to read one file, somewhere…
  - L. Malgeri, CMS physics coordinator: "It's like a dream come true!"
- …up to the entire CMS computing system
  - More efficient resource usage, more robust systems, more robust sites, easier to incorporate opportunistic resources
- We are just starting to understand its implications for the experiment, and for large-scale data management in general
  - LHC Run 2 will be a huge learning experience
- We're looking forward to future developments in this area